Building up Development and Design Capabilities in Software through the Creation of Interfaces between Users and Producers: The Case of the Mexican Firms

José Luis Sampedro

Universidad Autónoma Metropolitana, Xochimilco

Área Desarrollo Tecnológico. México, D. F.

Globelics Academy, Lisbon May 30th 2005

Content

1. Aim
2. Research problem and research questions
3. Analytical framework
4. Research strategy
5. Software and technological evolution
6. Structure about the Mexican software sector
7. Stylized facts

1. Aim

The aim of this work is to explain and analyze the process of construction of interfaces between users and producers firms of software and how, through the interfaces, they create knowledge and accumulate technological capabilities.

2. Research problem and research question

- Mexican firms have followed up the production of customer software *of the proprietary type*.

- However, there are firms that have begun to develop and design customer software *of the free type*. They have higher possibility to create knowledge and build up technological capabilities in an incremental way

- How do Mexican firms build up interfaces and why these are important in the creation of knowledge and in the process of technological capability accumulation?

  - What kind of interfaces do the Mexican firms create?

3. Analytical framework

Techno-Economic Paradigm (TEP):
"A change into the collective conscience that become the *common sense* of engineers, managers, investors, entrepreneurs…for obtain the maximum efficiency and the best-practice productive"
(Freeman and Perez, 1988; Perez, 1986, 2003)

- Structural change
- Technological revolution
- Technological system

Level 3

- Radical innovation
- Incremental innovation

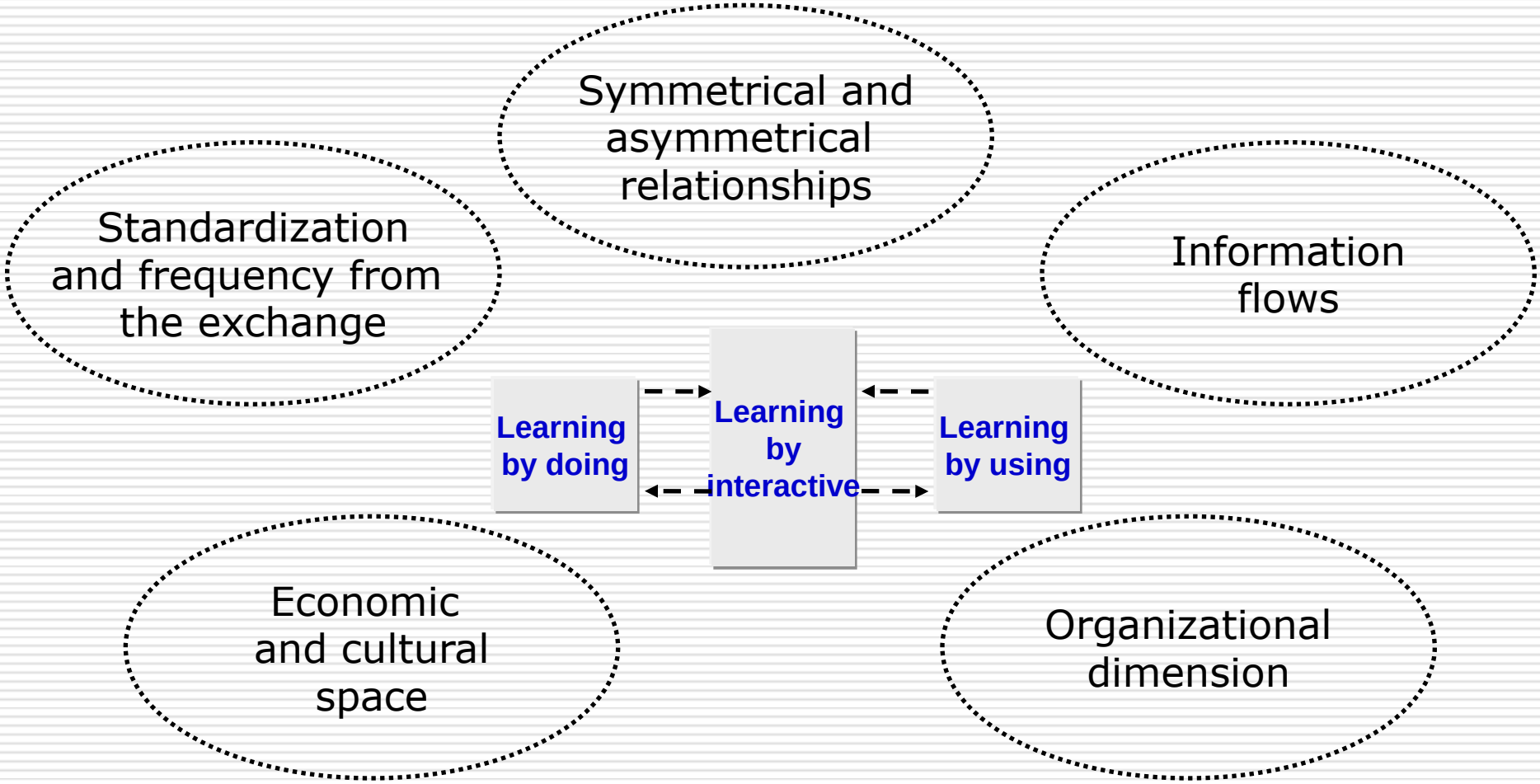
Level 2

Invention-Innovation-Diffusion

Level 1

User-Producer interaction in the process of innovation

(Lundvall, 1985, 1988, 1992)

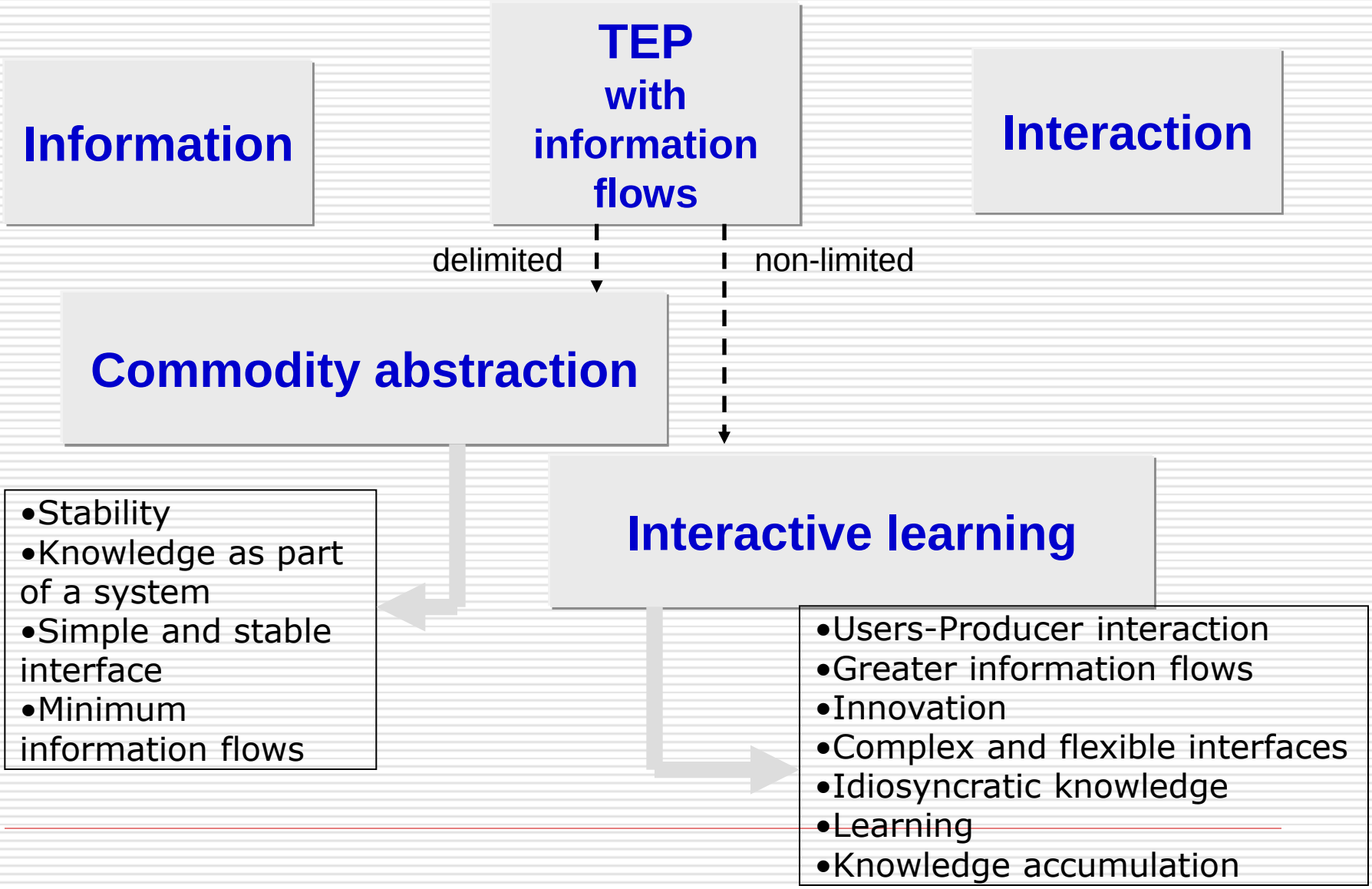


Interface and TEP (1)

(Andersen, 1991, 1996)

- **Interface can be defined as a coordination between users and producers who exchange different types of information. If a interface is accepted and stable, the information necessary of each agent will be delimited. But, the innovative process will presuppose an information-rich interaction and thereby often presuppose non-standardized interfaces but complex interfaces.**

Interface and TEP (2)



Interfaces and TEP (3)

Problems originated in the environment could be solved through the interface

Simple interface
(Commodity abstraction)

- Mature technology
- Difficult to do exchange
- Routines are established under this principle
- Short information flows

Complex interface
(Interactive learning)

- New technology
- Flexibility change
- Large information flows
- Learning process
- Knowledge generation

Knowledge as a key input of the interface (1)

(Nonanka y Takeuchi, 1994; Davenport y Prusak 1998; Senker y Faulkner, 1996; Malerba y Orsenigo, 1999)

Information

- Flow of messages: data are manipulated for decision-making
- Judgments and meanings

- **It depends of specific contexts and is relational**
- **It is created dynamically during the social interaction**

knowledge

- Beliefs
- Created starting from messages flows (information)
- ***"Knowledge is a fluid combination of experiences, values, contextual information and expert ideas that provide a structure to evaluate and incorporate new experiences and information"*** (Davenport y Prusak, 1998:5)

Knowledge as a key input of the interface (2)

Tacit knowledge

- Personal, it depends from the specific context
- Cognitive dimension (models, diagrams)
- Technical dimension (know how)
- Expertise and practice

Explicit knowledge

- Articulated
- Transfer through formal and systematic languages (codes)

Technological capabilities (1)

The learning process and the develop of internal capabilities allows to the firms to improve their productivity and their innovative process at product, process and organizational level a long the time

(Maxwell, 1981; Bell, 1984; Bell y Pavitt, 1995; Lall, 1992, 2000; Hobday, 1995, 2000, 2001; Dodgson, 1993)

Technological capabilities (2)

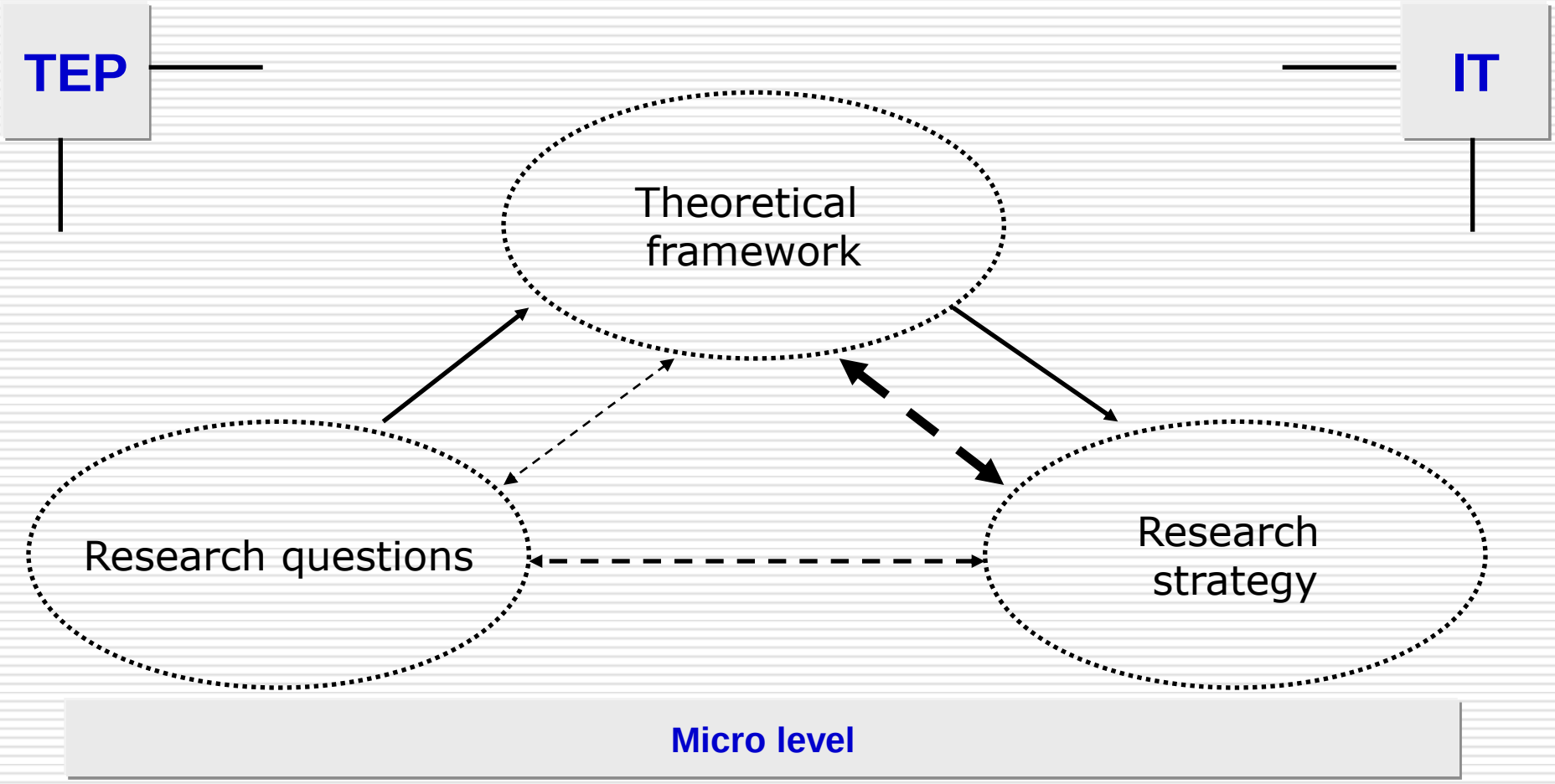
The basic idea is that capabilities represent abilities to do things, and technological capabilities reflect the mastering of technological activities

(Dutrénit, 2000; Dutrénit y Vera-Cruz, 2001; Vera-Cruz, 2003

Analytical category

Concept	Category
Interface	➢**Class of interfaces** ➢**Technological cooperation** ➢**Interactive process (user-producer linkage)**
Knowledge	➢**Generation of knowledge** ➢**Relevant knowledge** ➢**Sources of knowledge** ➢**Diffusion of knowledge**
Technological capabilities	➢**Design of software** ➢**Developed of software** ➢**Activities of learning**

4. Research strategy



Design research

- Research questions
- Theoretical proposal
- Unit of analysis (multiple-units of analysis)
- Logic linking between data and theoretical proposal
- Evaluation of the results

Research questions

1. How do Mexican firms build up interfaces and why these are important in the creation of knowledge and in the process of technological capability accumulation?

   What kind of interfaces do the Mexican firms create?

2. What kind of knowledge (tacit or explicit) is dominant in the process of creation of interfaces and in the develop and design process?

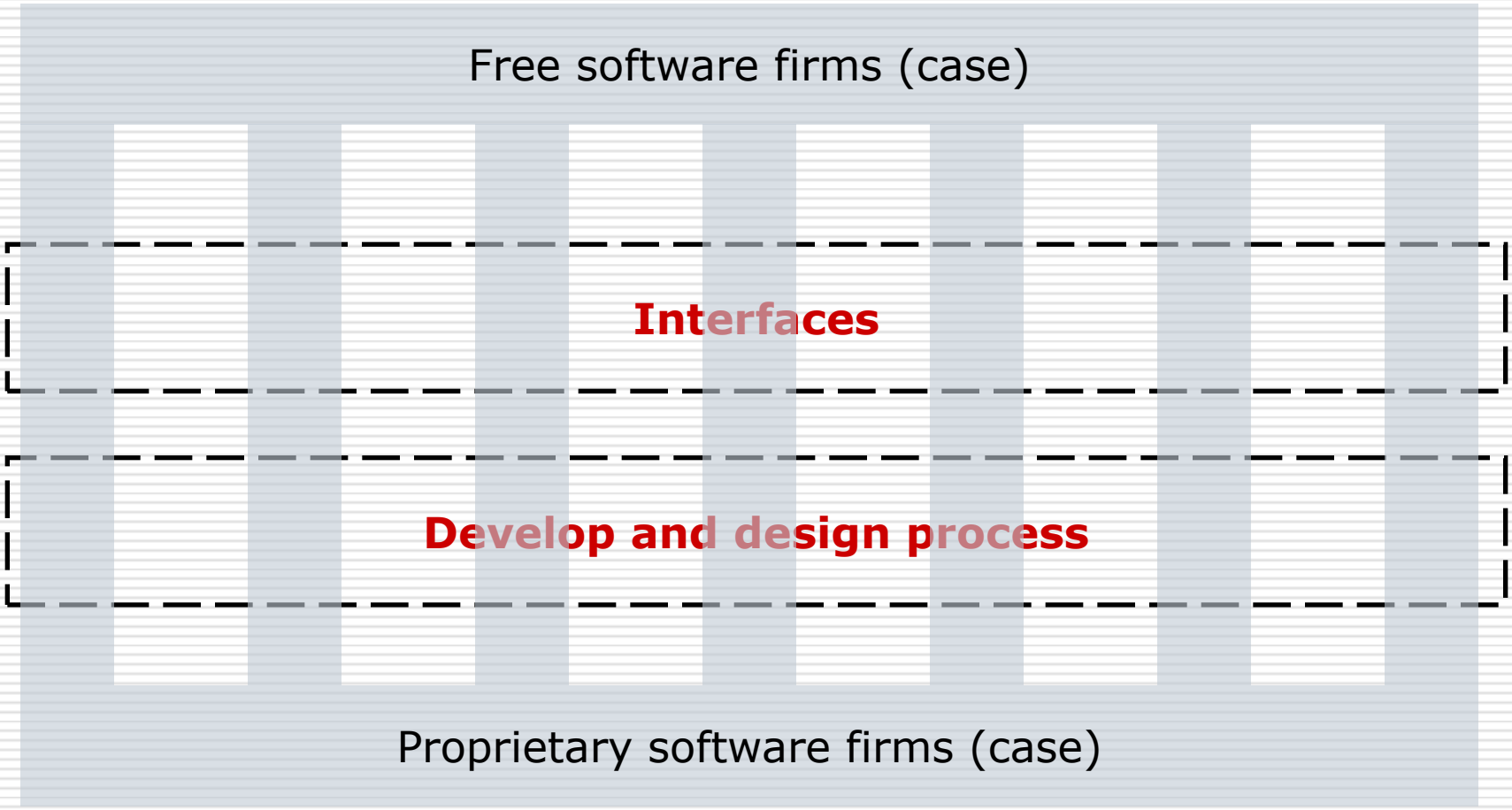
3. What importance does the develop and design process have in the generation of knowledge and in the process of technological capability accumulation?

   Are there differences between proprietary and free context?

Theoretical proposal

The interfaces are a crucial element in the process of generation of knowledge and in the process of technological capability accumulation, where the tacit knowledge is relevant. Unlike the firms that develop and design proprietary software, the firms that develop and design free software have a window of better opportunity to generate new knowledge and to accumulate technological capabilities through the creation of interfaces.

Units of analysis



Exploratory multiple-case study

Sources of information

Sources of information	**Detail**
(1) Open interview (17 interviewed)	To leaders of project, and developers of software
(2) Direct observation	Imply direct observation about the develop and design of software
(3) Internal and external documents	Books, magazines, papers, files

5. Software and technological evolution

- Software:
  - Technical: algorithm involves natural and social process
  - Link between the society and technology
  - Root: the code
  - Principal input: knowledge
  - Knowledge that is tacit, indefinite, dynamic…always in evolution
- Customer software: application done in base from the requirements of the users, is specific at sector and firm level

Evolutive path of the computers, 1950's-1990's

Mainframes	•1940's: rustic software, scientific and military •Supply for computer producers •1950's: industrial using
Minicomputers	•1965, industrial introduction in small firms •Software supply for computer producers •Specialized software •IBM separated the prizes of software from hardware
Microcomputers	•1970's: micro processor, •Operative systems •Industrial applications
Works stations	•1981-82 •CAD, CAE
PDA	•Personal digital assistant •Personal mobile assistant

Evolutive path of the software, 1950's-1990's

1940-1950, software development by computer producers

Specific software and only for each user

Cooperation from users into the development for better applications

1964, 360/IBM: software compatible over different computers

Mass applications

Software develops by independents firms

Growth	Consolidation
1970's: 1500 independent software firms	Dominant design for PC,
1971: IBM introduces the 'hard disk'	Down prizes of hardware
Package software for PC's	
Problems with quality, management, measure, prize	Operative systems and application for PC are homogeneous

Networks	Free software
1980’s, work in networks: Intranets e Internet based on PC	Richard Stallman, 1980’s
1990’, software determinate by work in networks, work in teams	Software develops into virtual networks
Necessity for development customer software	Open source code

Rationality

To improve the quality and the effectiveness
from the development

Process replication of development

Had been possible to manage
but not to rationalize

Varieties of Software	Class of software		
	Operative Systems	**Tools**	Solutions (programs)
Package Software	* o	o *	o *
Specialized Software	o *	o *	o *
Customer Software	o *	o *	o *

o Free Software, * Proprietary Software
Source: Own elaboration.

6. Structure about the Mexican software sector

- 1992-2001
  - Software: 0.1 / GNP
  - TI: 1% / GNP
  - Software: 7.2% / TI national
  - Hardware: 37% / TI national

1000 – 1500 proprietary software firms

100 free software firms, 1000 developers and 10 000 users

Proprietary software segment (1)

Concept	Employment	Average employment	Firms Num.
Micro	< 15	7	63
Small	De 16 a 100	60	117
Medium	De 101 a 250	175	14
Large	De 251 a 1,000	600	11
Corporative	> 1,000	1,500	1

Fuente: AMITI, 2001

Proprietary software segment (2)

- 87% are small and medium size
  - < 60 L / firm
- 6.7% are medium size
- 5.3 large
- One corporative firms, 1500 employees
- Competitive firms into the international markets:
  - 250 L, and have to grown to 1000
  - CCMi (AMITI, 2001)

Free software segment

- 100 enterprises
  - 90% micro-small, < 15 employees
- 1000 developers and 10 000 users
- 2002, 7.9 Free Software / Proprietary Software
- 2004, 9.0 Free Software / Proprietary Software
- Niche: servers
  - 2003: 15 000
  - 2004: 20 000

Mexican software segment

- 2006, sales: $5,000.00 mdd
  - 1.5% total annual production
  - To create 100 000 local employees
  - Demand of IT: services, electronic, government, financial sectors, manufacture

7. Stylized facts

Proprietary context

- Interfaces tend to be simples and standardizes
- Knowledge generated inside of the firms
- Software develops in the context of industrial secret

Free context

- Interfaces tend to be complex, non-standardizes
- Knowledge generated into ‘communities of developers’
- Software develops in the context of ‘free’ and work in teams (networks)

Proprietary software context

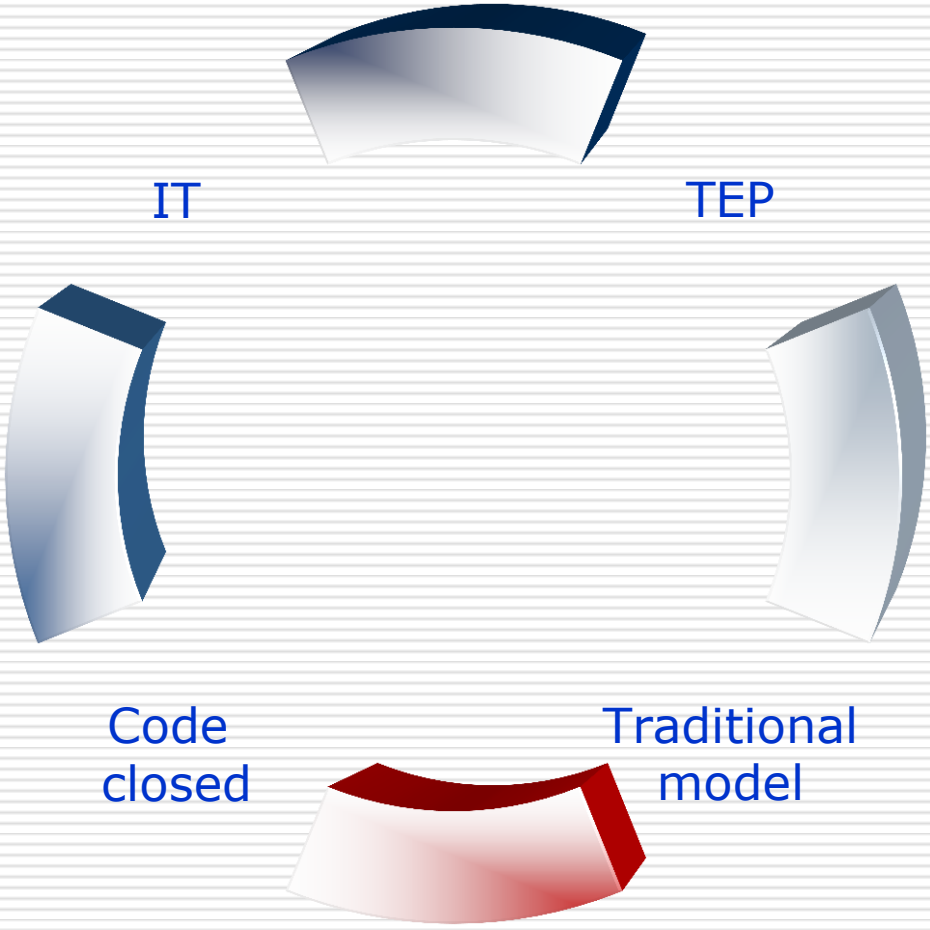
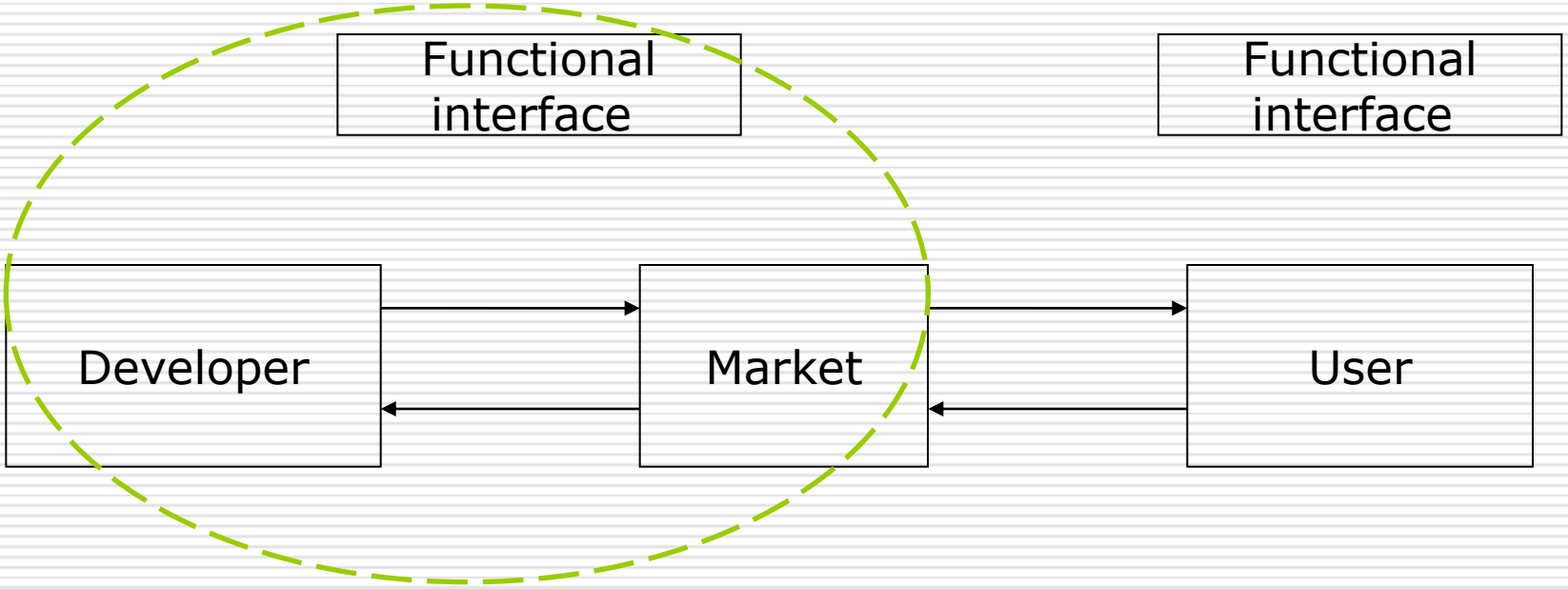


Diagram of a interface. Proprietary software



Free software context

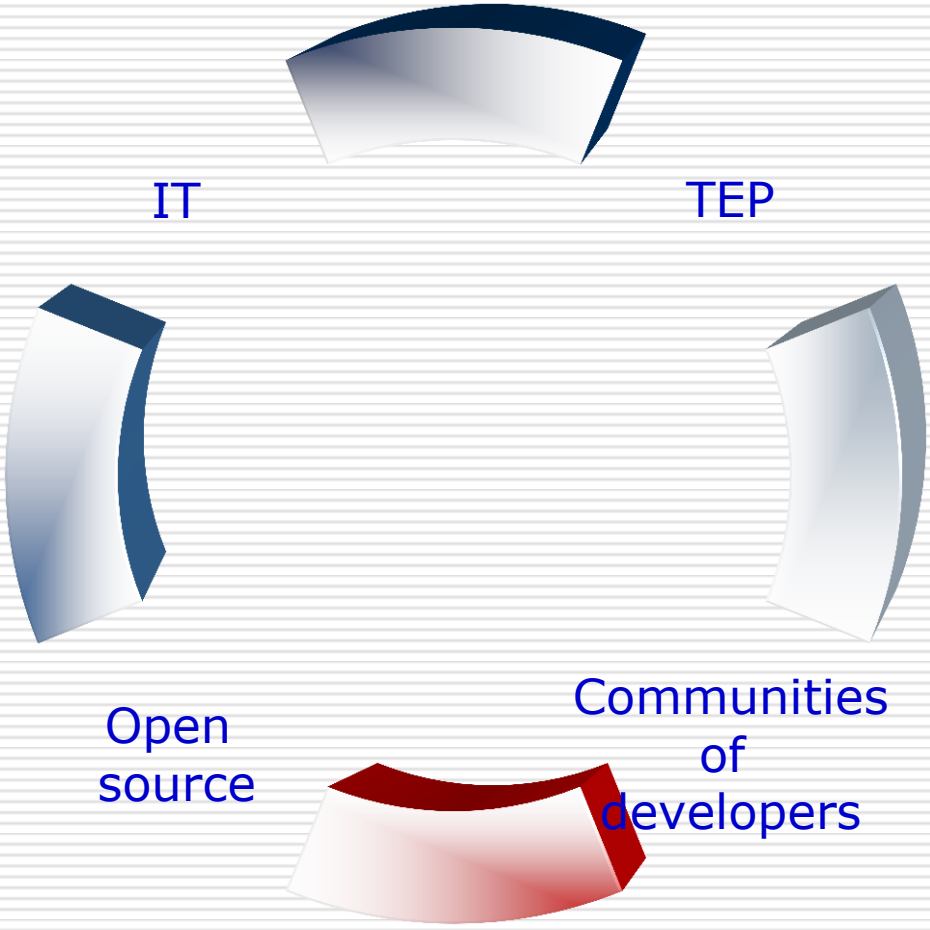
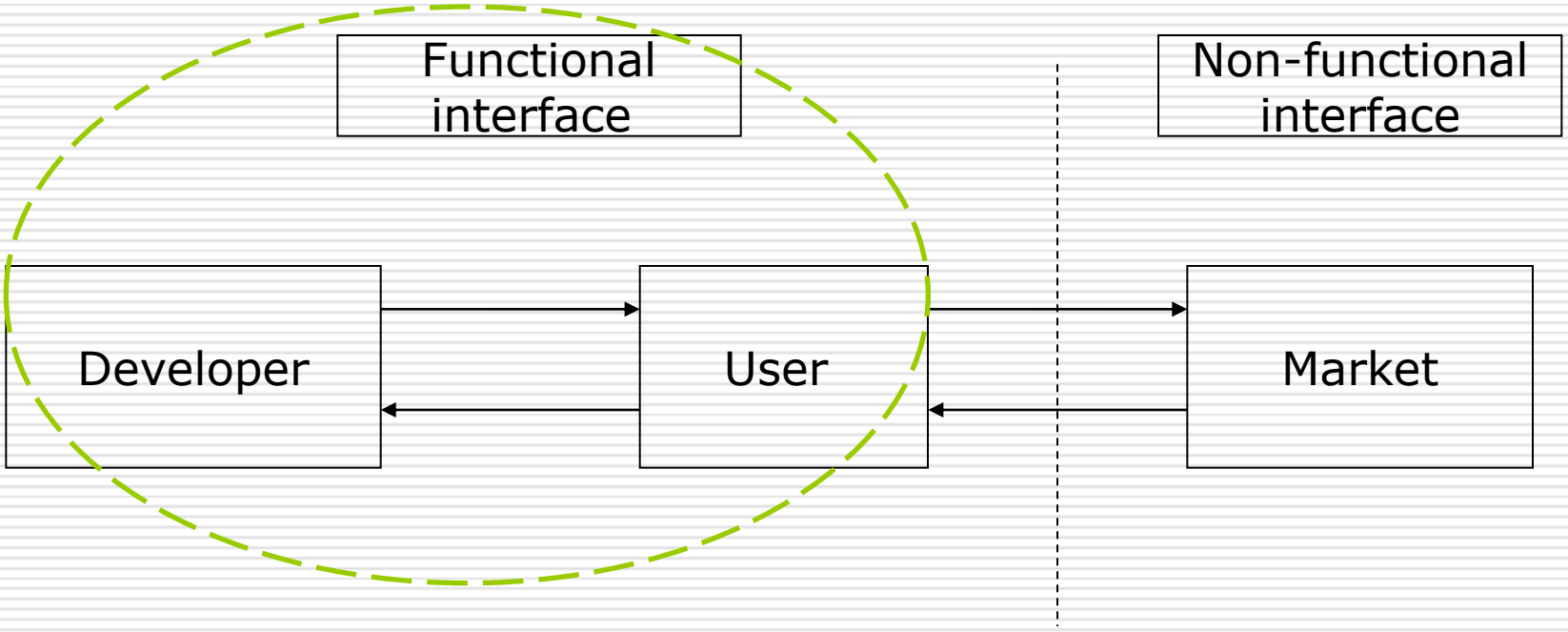


Diagram of a interface. Free software



Toward a taxonomy of interface-knowledge-capabilities

Concept	Free software	Proprietary software
Interface	Knowledge (complex)	Market (stable)
	•Organizational (stable) •Productive (complex) (embedded software)	
Knowledge	Generated into communities of developers	Generated inside of the firm
Technological capabilities	•Traditional design •Development into the communities	•Traditional design •Traditional development

Source: Own elaboration.

Interfaces: there are types and depths in each context (free and proprietary) that depend of:

- **Proprietary context creates traditional and stables interfaces**
- **Free context creates complex and non-standardized interfaces**

- **Knowledge: predominant information flows on knowledge flows**